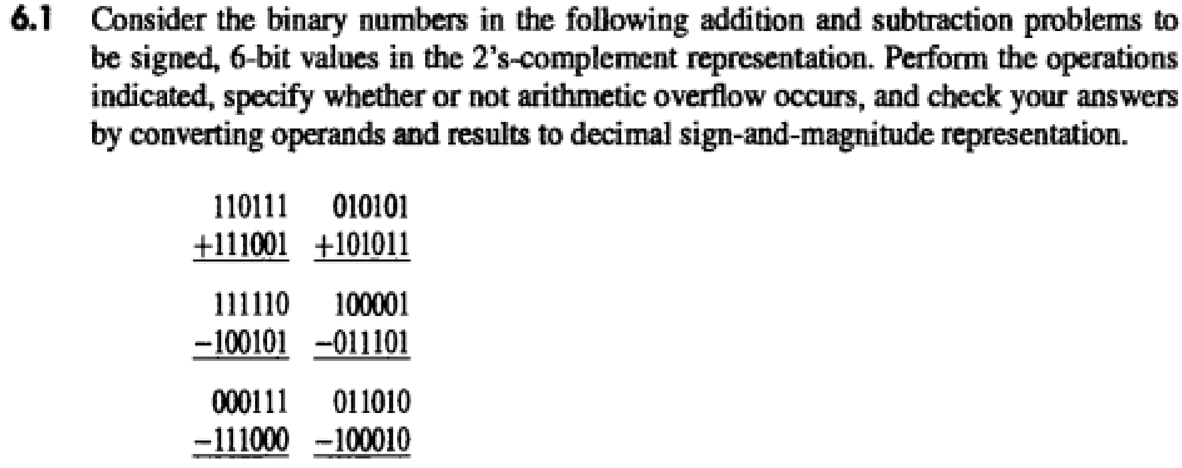
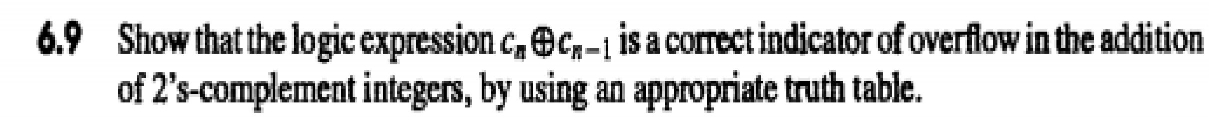
Computer Architecture

Sheet (6)

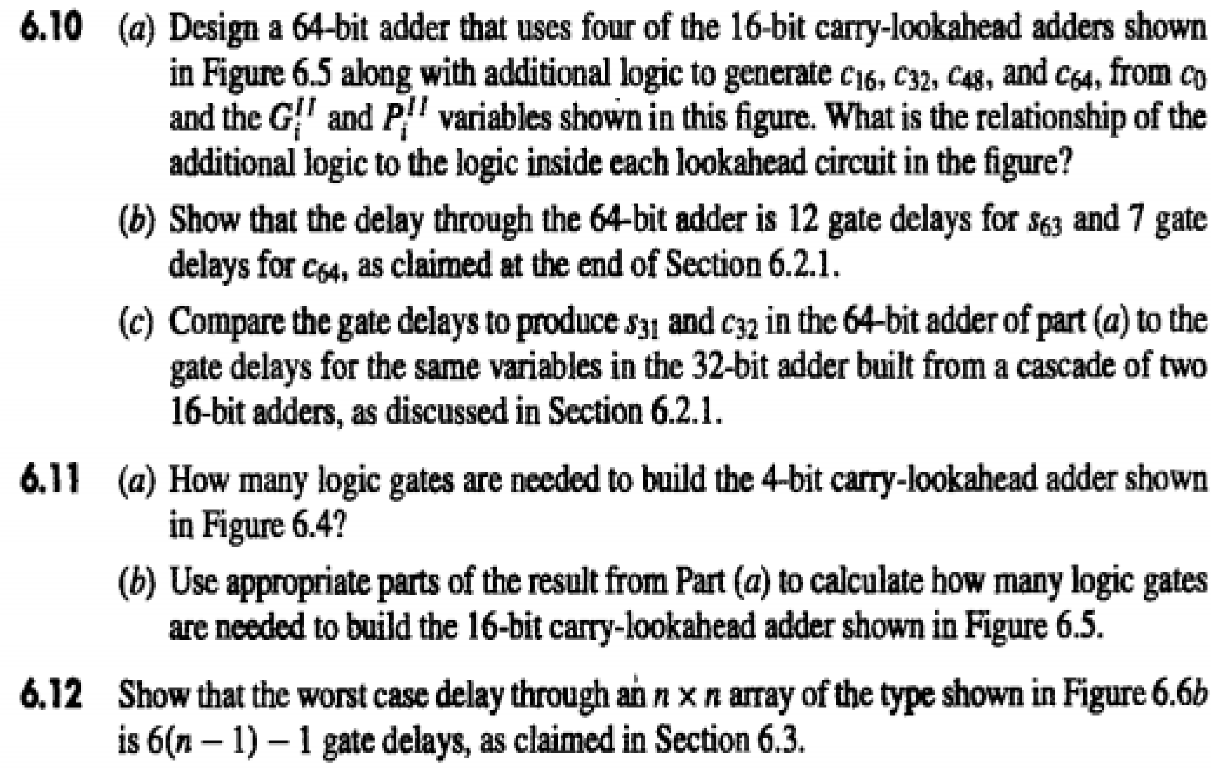


6.1 Consider the binary numbers in the following addition and subtraction problems to be signed, 6-bit values in the 2's-complement representation. Perform the operations indicated, specify whether or not arithmetic overflow occurs, and check your answers by converting operands and results to decimal sign-and-magnitude representation.

$$\begin{array}{rr} 110111 & 010101 \\ \underline{+111001} & \underline{+101011} \\ \\ 111110 & 100001 \\ \underline{-100101} & \underline{-011101} \\ \\ 000111 & 011010 \\ \underline{-111000} & \underline{-100010} \end{array}$$



6.9 Show that the logic expression $c_n \oplus c_{n-1}$ is a correct indicator of overflow in the addition of 2's-complement integers, by using an appropriate truth table.



6.10 (*a*) Design a 64-bit adder that uses four of the 16-bit carry-lookahead adders shown in Figure 6.5 along with additional logic to generate c_{16}, c_{32}, c_{48}, and c_{64}, from c_0 and the G_i^{II} and P_i^{II} variables shown in this figure. What is the relationship of the additional logic to the logic inside each lookahead circuit in the figure?

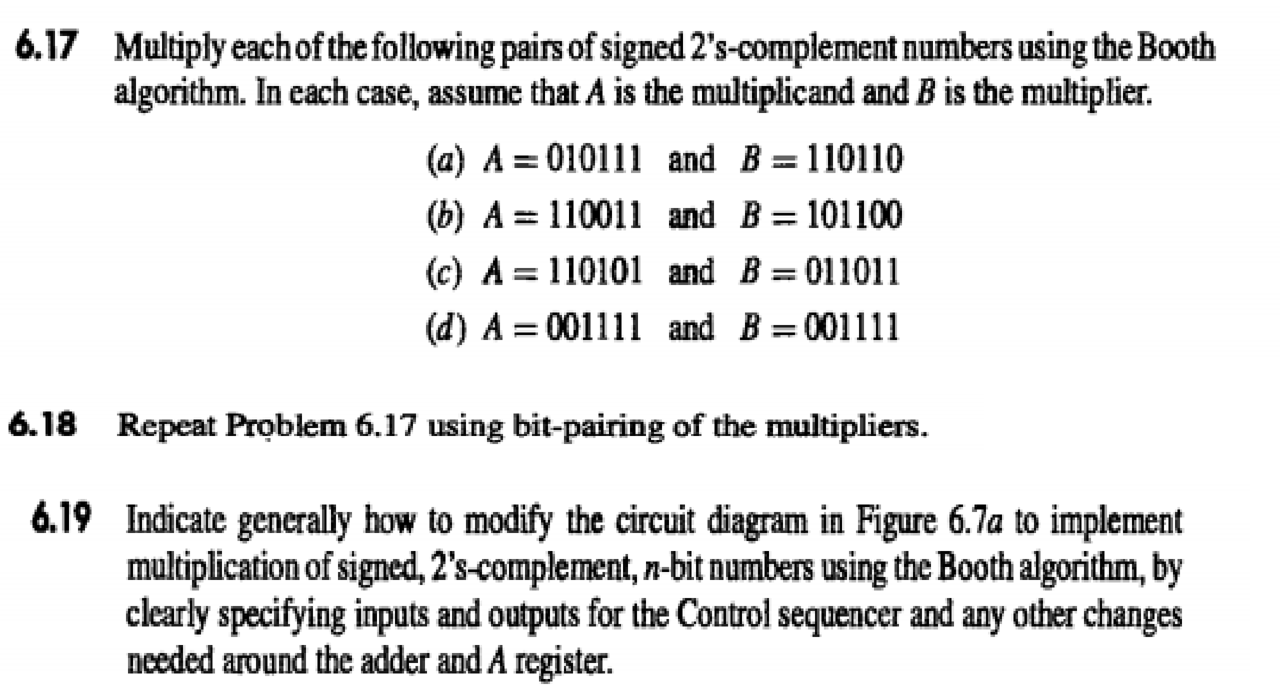
(*b*) Show that the delay through the 64-bit adder is 12 gate delays for s_{63} and 7 gate delays for c_{64}, as claimed at the end of Section 6.2.1.

(*c*) Compare the gate delays to produce s_{31} and c_{32} in the 64-bit adder of part (*a*) to the gate delays for the same variables in the 32-bit adder built from a cascade of two 16-bit adders, as discussed in Section 6.2.1.

6.11 (*a*) How many logic gates are needed to build the 4-bit carry-lookahead adder shown in Figure 6.4?

(*b*) Use appropriate parts of the result from Part (*a*) to calculate how many logic gates are needed to build the 16-bit carry-lookahead adder shown in Figure 6.5.

6.12 Show that the worst case delay through an $n \times n$ array of the type shown in Figure 6.6*b* is $6(n - 1) - 1$ gate delays, as claimed in Section 6.3.



6.17 Multiply each of the following pairs of signed 2's-complement numbers using the Booth algorithm. In each case, assume that A is the multiplicand and B is the multiplier.

(*a*) $A = 010111$ and $B = 110110$
(*b*) $A = 110011$ and $B = 101100$
(*c*) $A = 110101$ and $B = 011011$
(*d*) $A = 001111$ and $B = 001111$

6.18 Repeat Problem 6.17 using bit-pairing of the multipliers.

6.19 Indicate generally how to modify the circuit diagram in Figure 6.7*a* to implement multiplication of signed, 2's-complement, n-bit numbers using the Booth algorithm, by clearly specifying inputs and outputs for the Control sequencer and any other changes needed around the adder and A register.

***) Using non-restoring division perform the operation A % B on the five bit numbers A=10101 and B=00101.**